# P&S Modern SSDs

## Cutting-Edge Research in SSDs

Dr. Jisung Park
Prof. Onur Mutlu
ETH Zürich
Fall 2022
04 January 2023

# Access Control-Based Data Sanitization

## Evanesco: Architectural Support for Efficient Data Sanitization in Modern Flash-Based Storage Systems


Myungsuk Kim*
morssola75@davinci.snu.ac.kr
Seoul National University

Jisung Park*
jisung.park@inf.ethz.ch
ETH Zürich & Seoul National University

Geonhee Cho
ghcho@davinci.snu.ac.kr
Seoul National University

Yoona Kim
yoonakim@davinci.snu.ac.kr
Seoul National University

Lois Orosa
lois.orosa@inf.ethz.ch
ETH Zürich

Onur Mutlu
omutlu@gmail.com
ETH Zürich

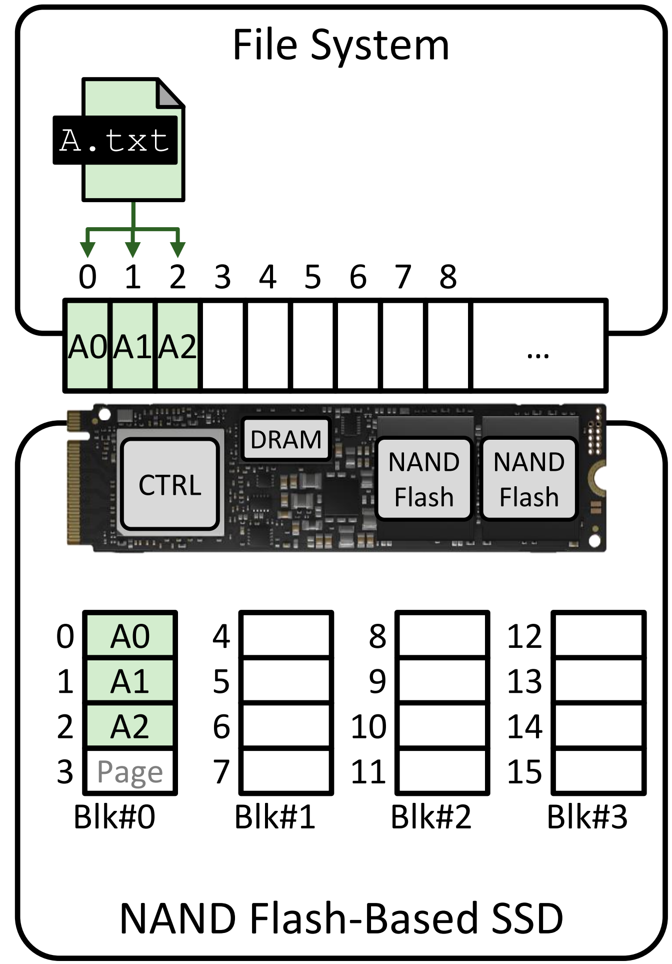
Jihong Kim
jihong@davinci.snu.ac.kr
Seoul National University

*The first two authors contributed equally to this research.


The 25th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS'20)

# Data-Remanence Problem in SSDs

- Deleted data can remain in SSDs for indefinite time

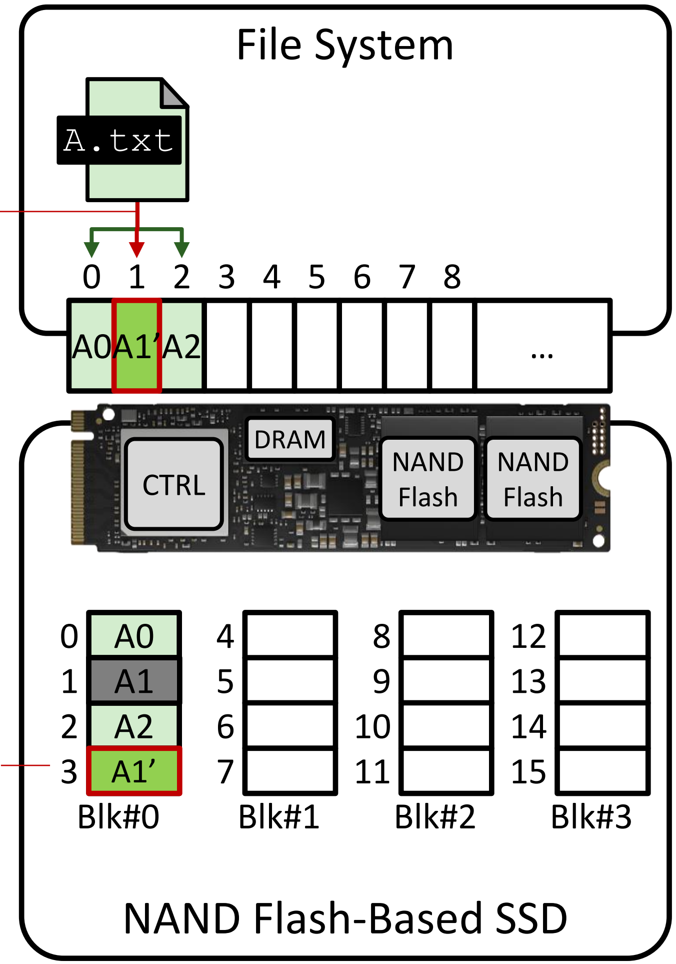


# Data-Remanence Problem in SSDs

- Deleted data can remain in SSDs for indefinite time

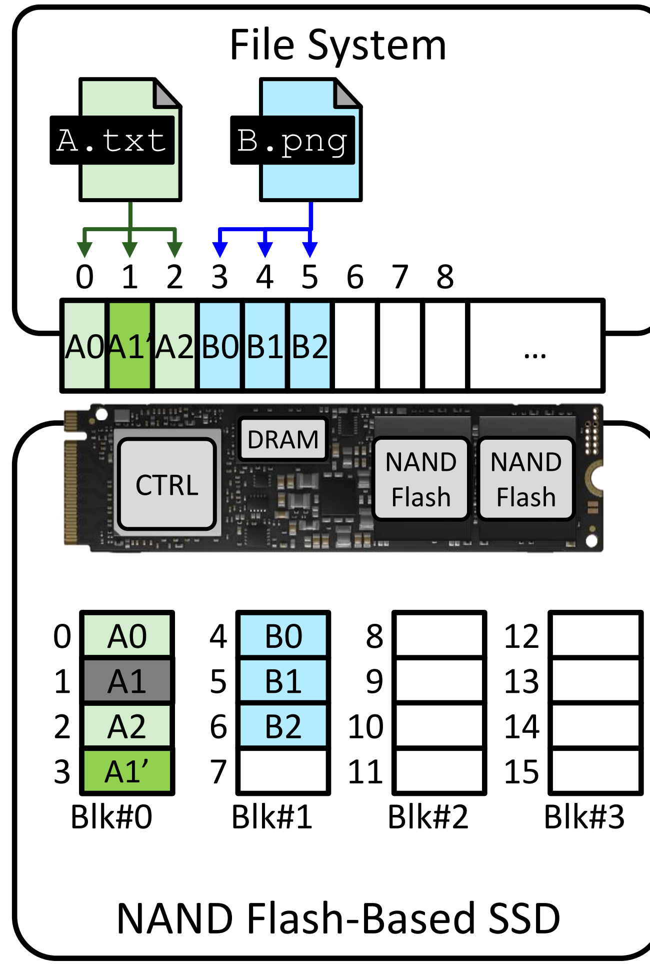
Update

No overwrite
→ Out-of-place update



# Data-Remanence Problem in SSDs

- Deleted data can remain in SSDs for indefinite time

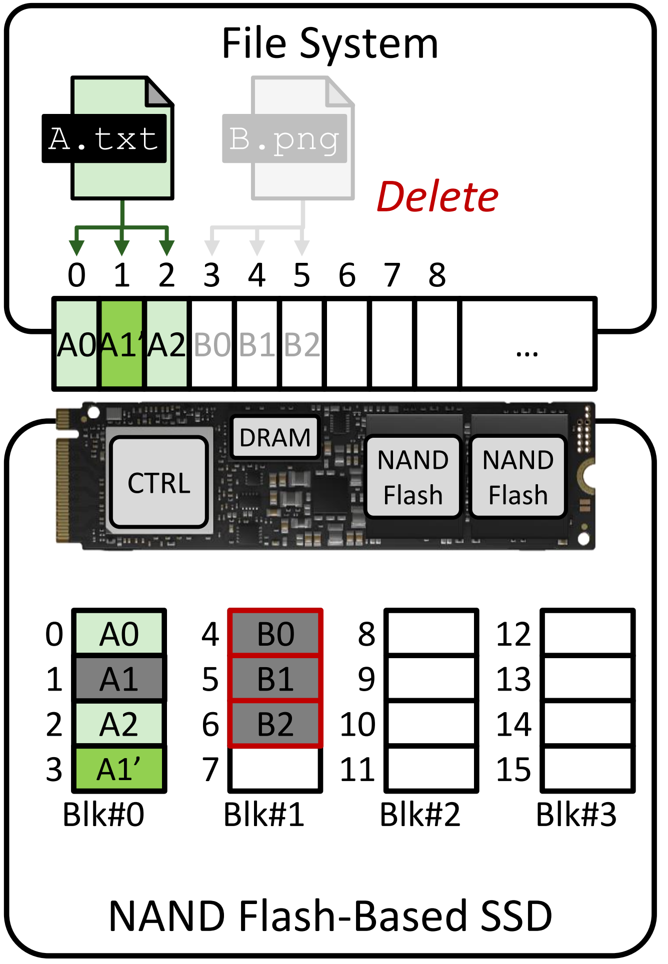


# Data-Remanence Problem in SSDs

- Deleted data can remain in SSDs for indefinite time

Q: When is a page erased?

A: Only in garbage collection
= when running out of free pages



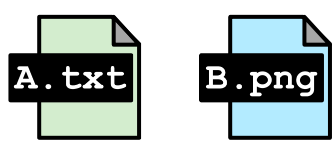
# Data-Recovery Attack

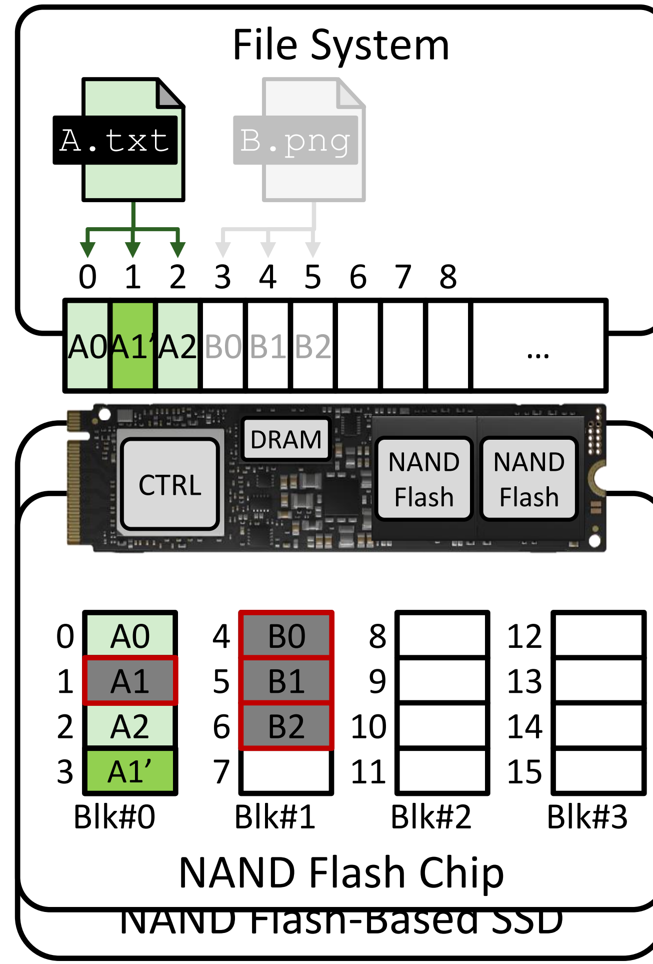
- System requirement: Obsolete data must be inaccessible


1. Detach the SSD
2. Detach the chip
3. Direct access to the chip

| A0 | A1 | A2 | B1 |
|---|---|---|---|

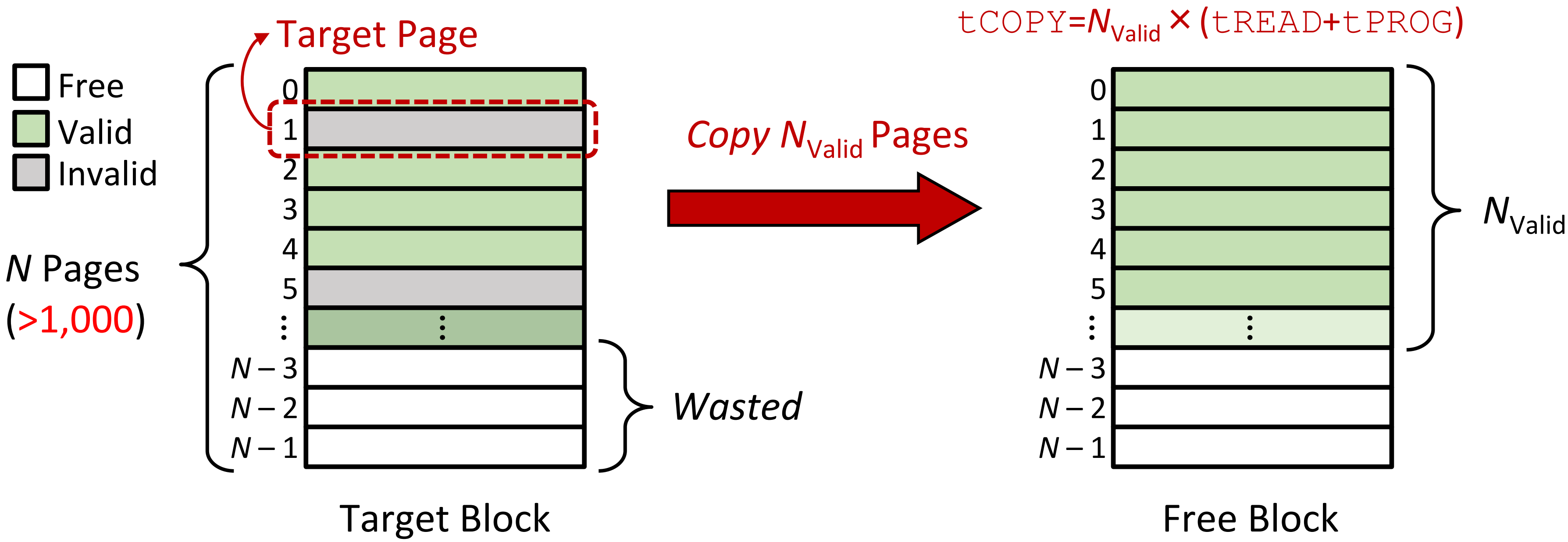
4. Run forensic tools



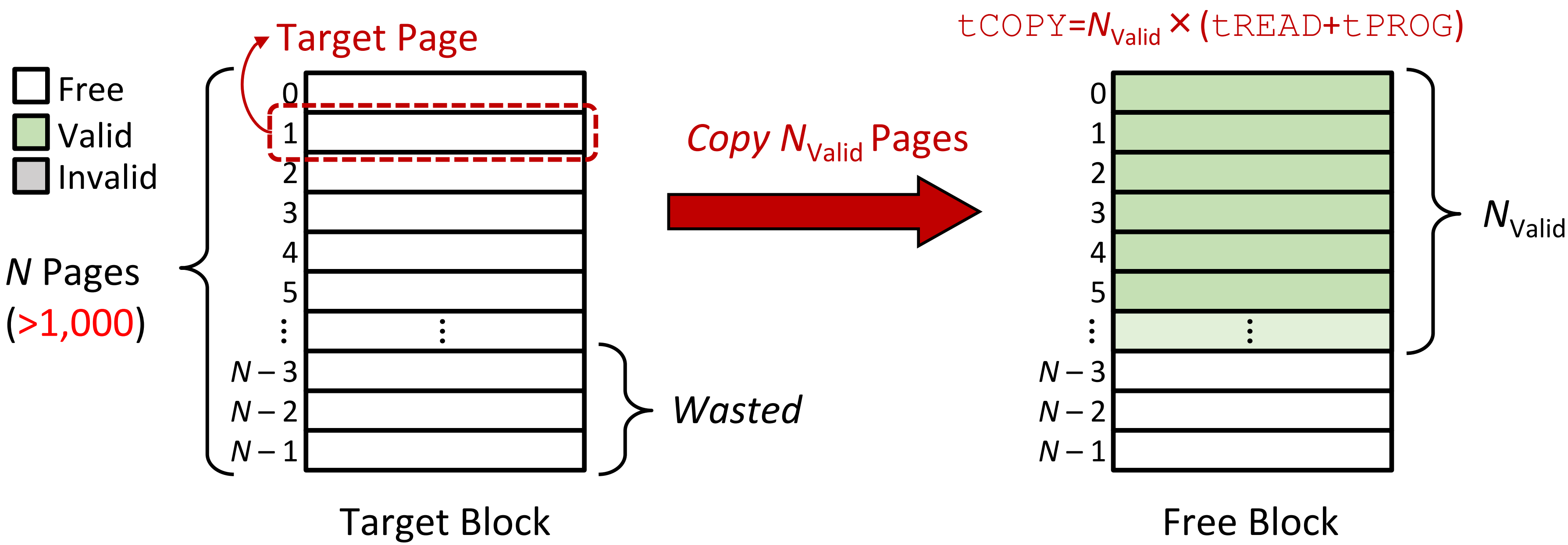
# Existing Solutions

- Why not immediately erase an invalid page?
  - Erase unit: a block (> 1,000 pages)



# Existing Solutions

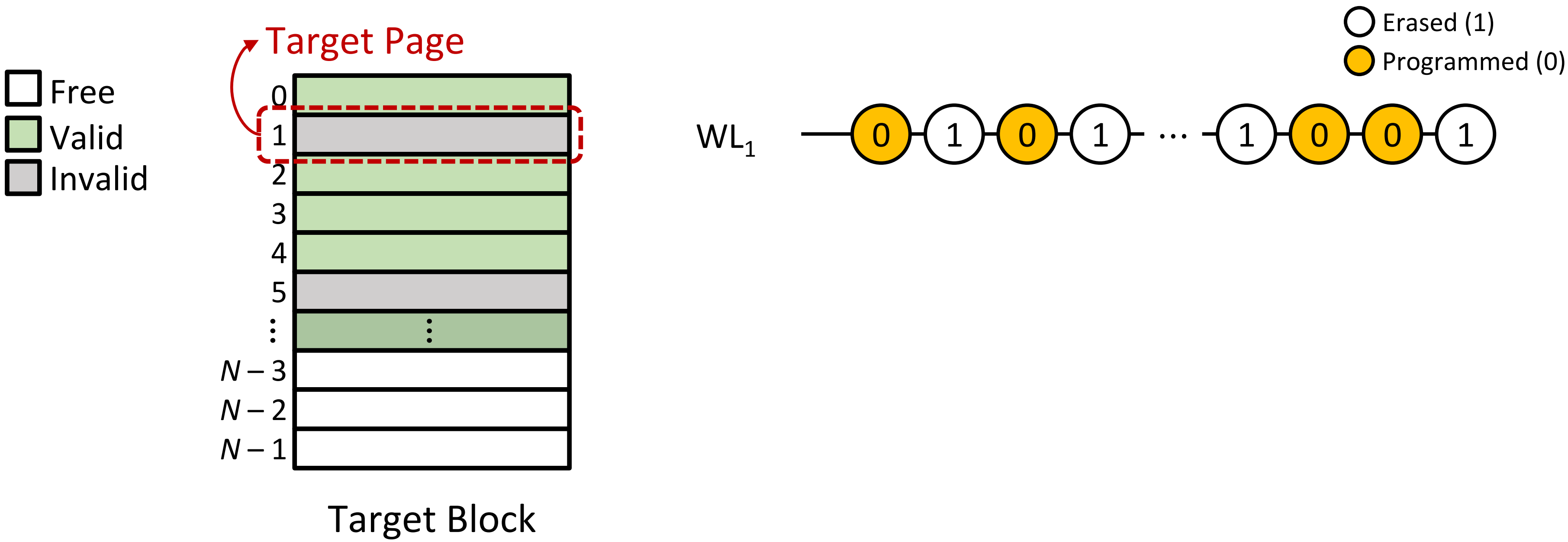
- Why not immediately erase an invalid page?
  - Erase unit: a block (> 1,000 pages)




Immediate block erasure causes prohibitive performance and lifetime overhead

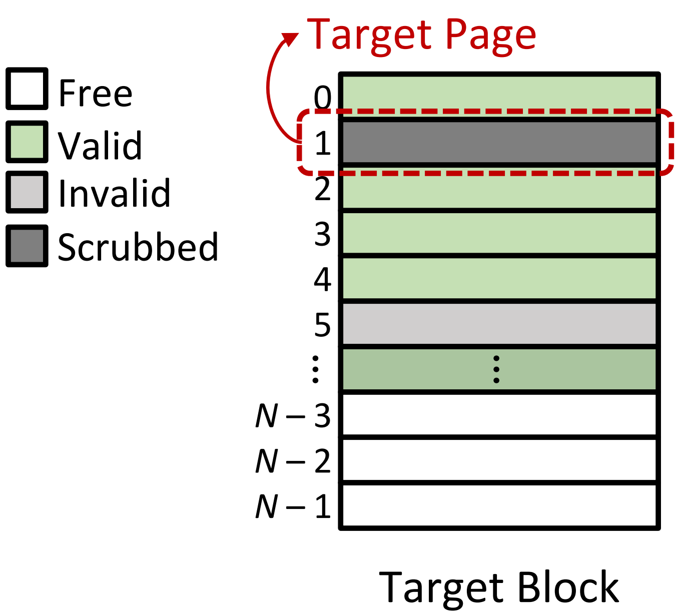
# Existing Solutions

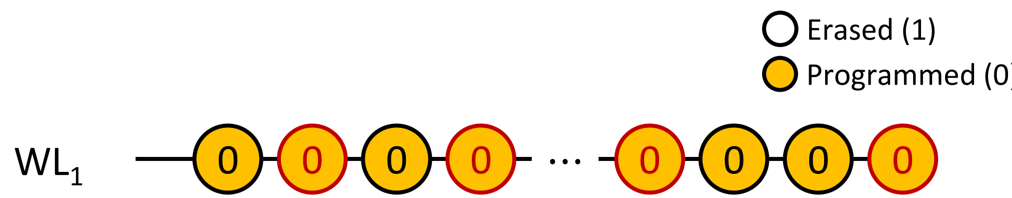
- Scrubbing [Wei+, FAST'11]
  - Reprograms all the flash cells storing an invalid page



# Existing Solutions

- Scrubbing [Wei+, FAST'11]
  - Reprograms all the flash cells storing an invalid page




*Scrubbing: Reprogram the WL (to all '0's)*

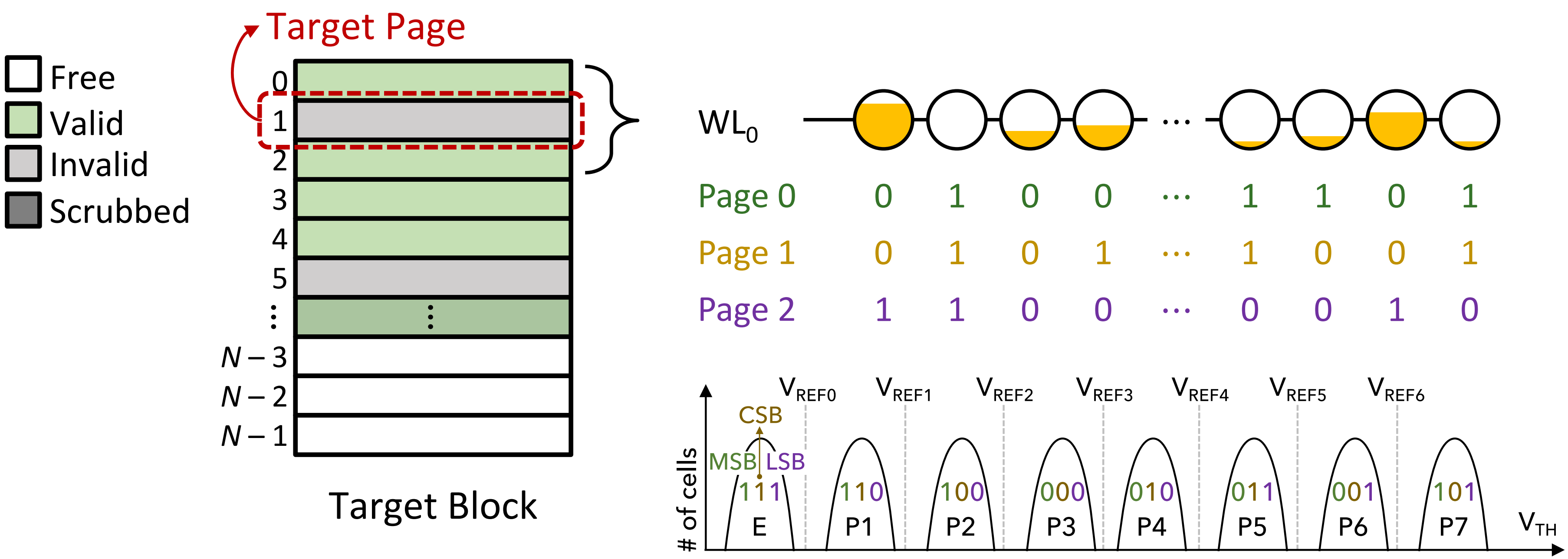
*Overwriting is infeasible usually, except when the data is all '0's*

*Destroys the page data w/o block erasure*

Problem solved?

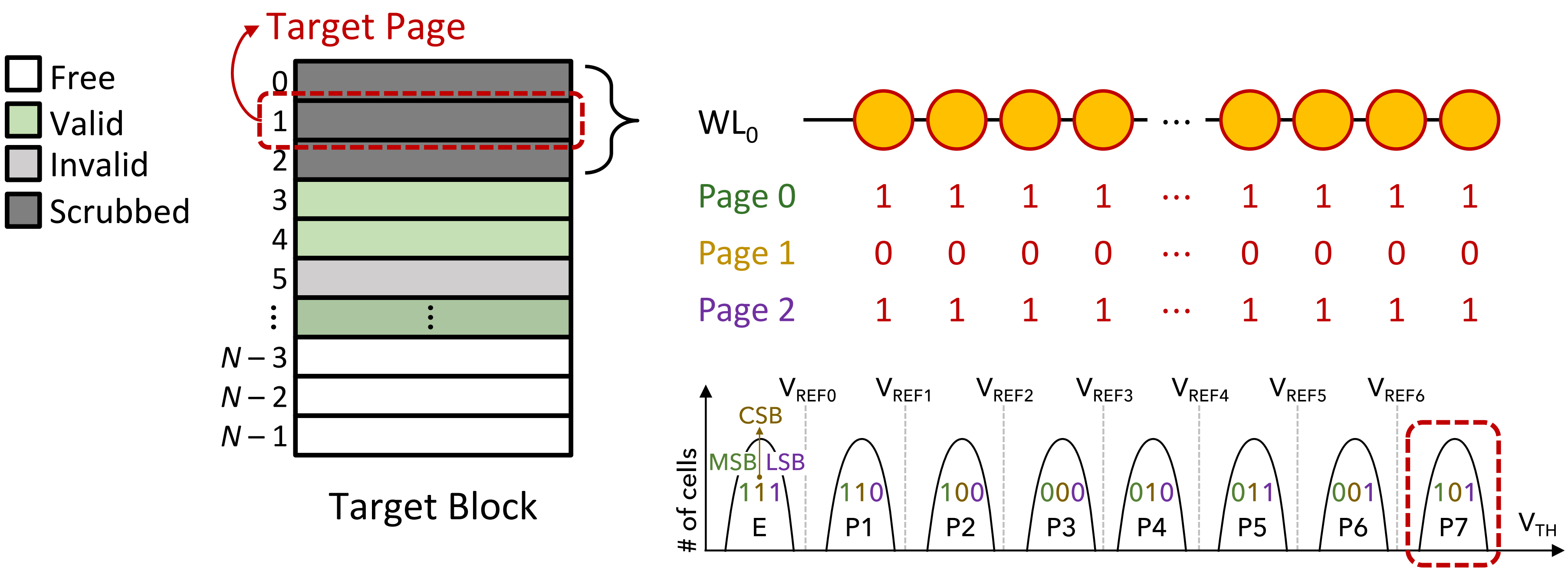
# Existing Solutions

- Scrubbing [Wei+, FAST'11]
  - Problem1: MLC NAND flash stores multiple pages in a WL



# Existing Solutions

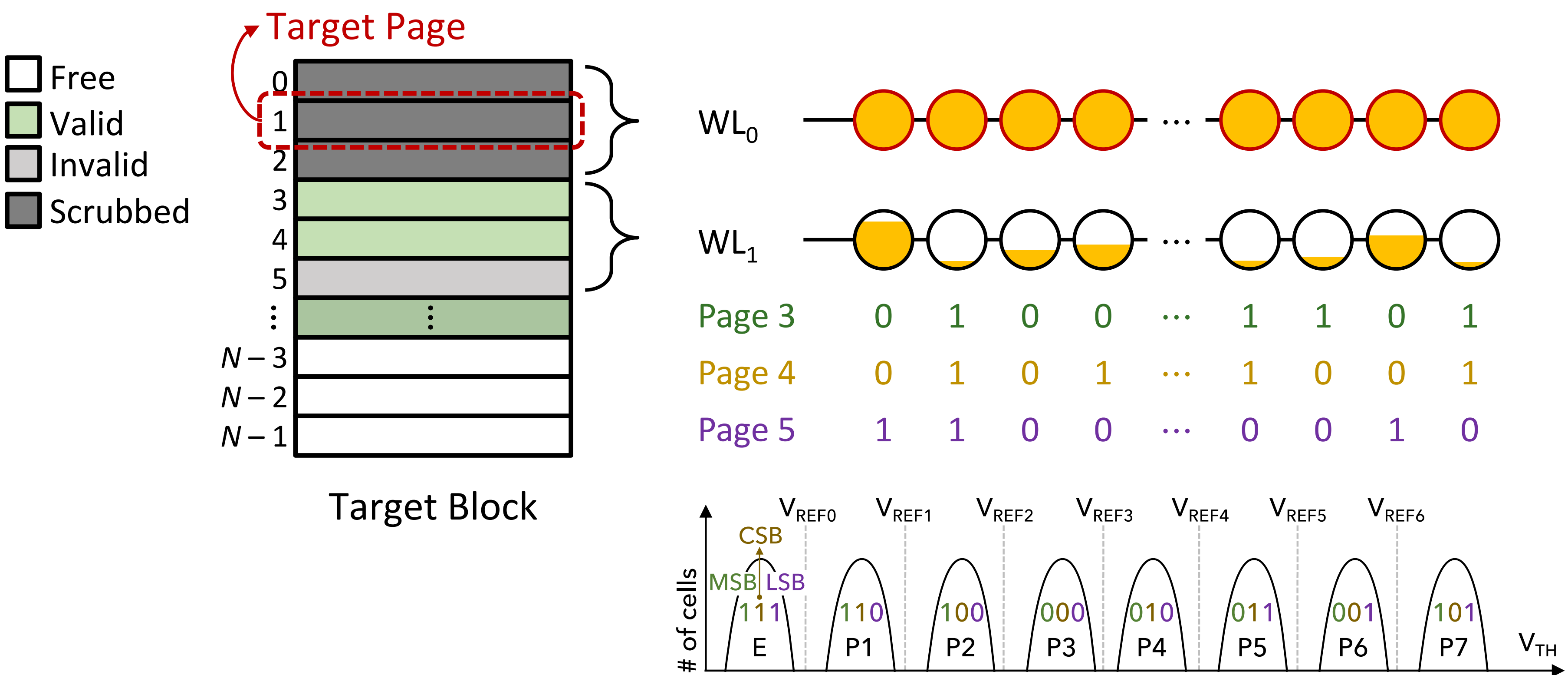
- Scrubbing [Wei+, FAST'11]
  - Problem1: MLC NAND flash stores multiple pages in a WL




Scrubbing in MLC NAND flash memory
→ Destroys other valid pages → Copy overheads

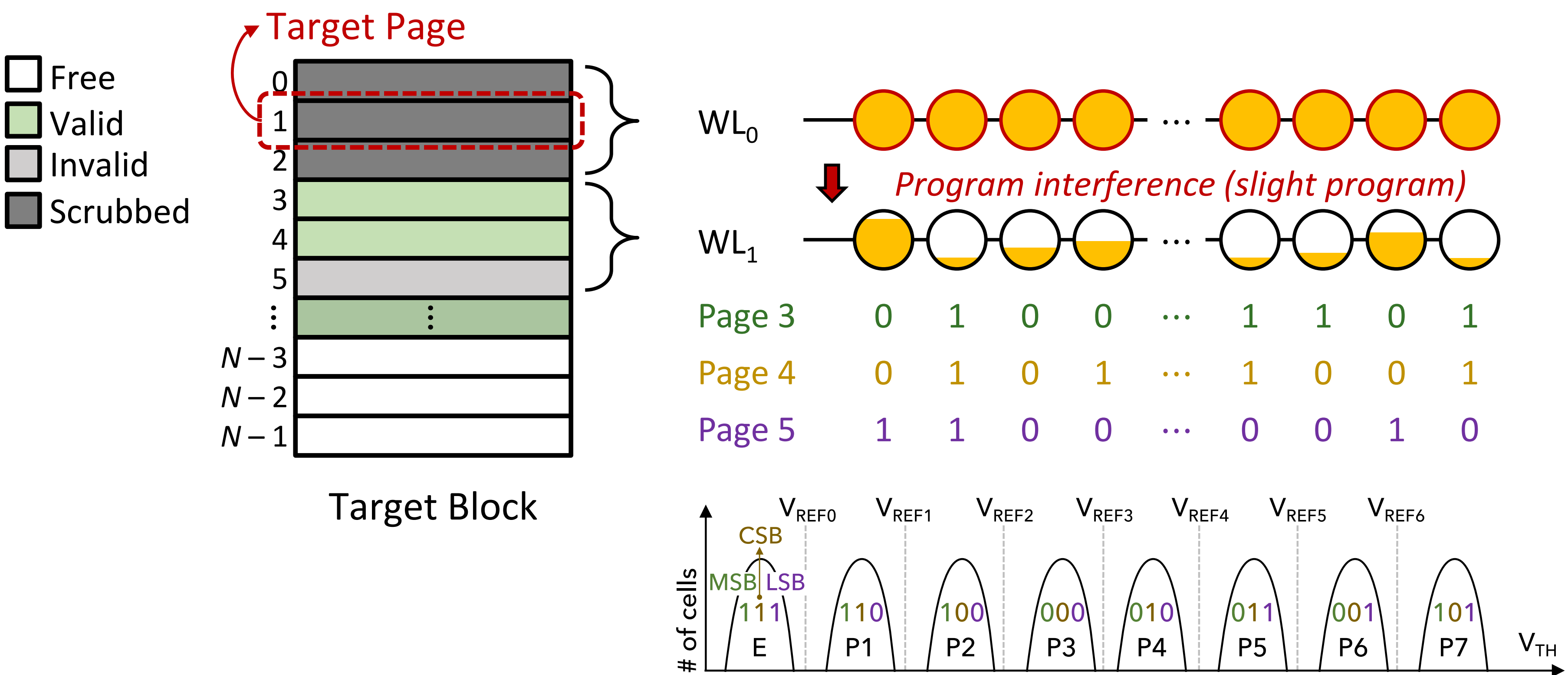
# Existing Solutions

- Scrubbing [Wei+, FAST'11]
  - Problem 1: MLC NAND flash stores multiple pages in a WL
  - Problem 2: Program interference



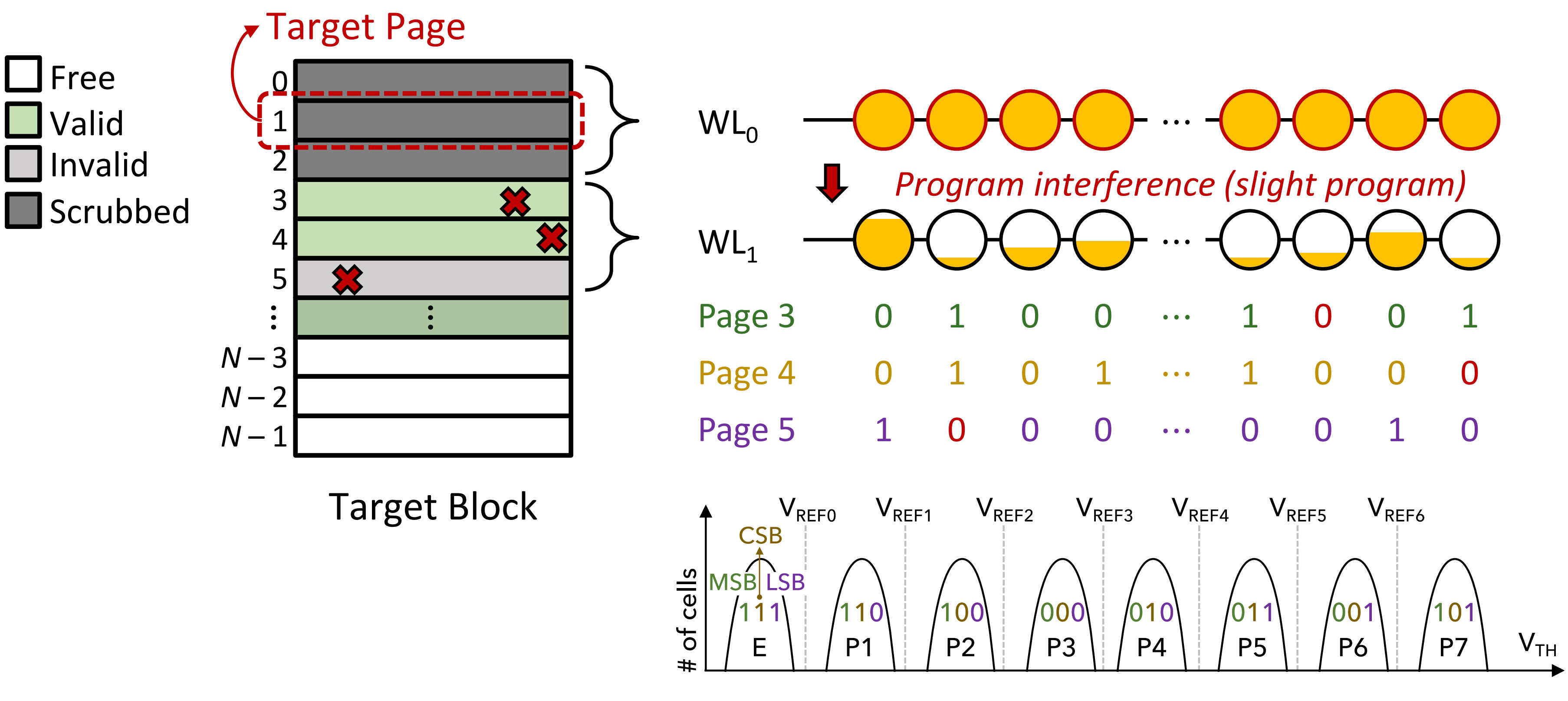
# Existing Solutions

- Scrubbing [Wei+, FAST'11]
  - Problem 1: MLC NAND flash stores multiple pages in a WL
  - Problem 2: Program interference



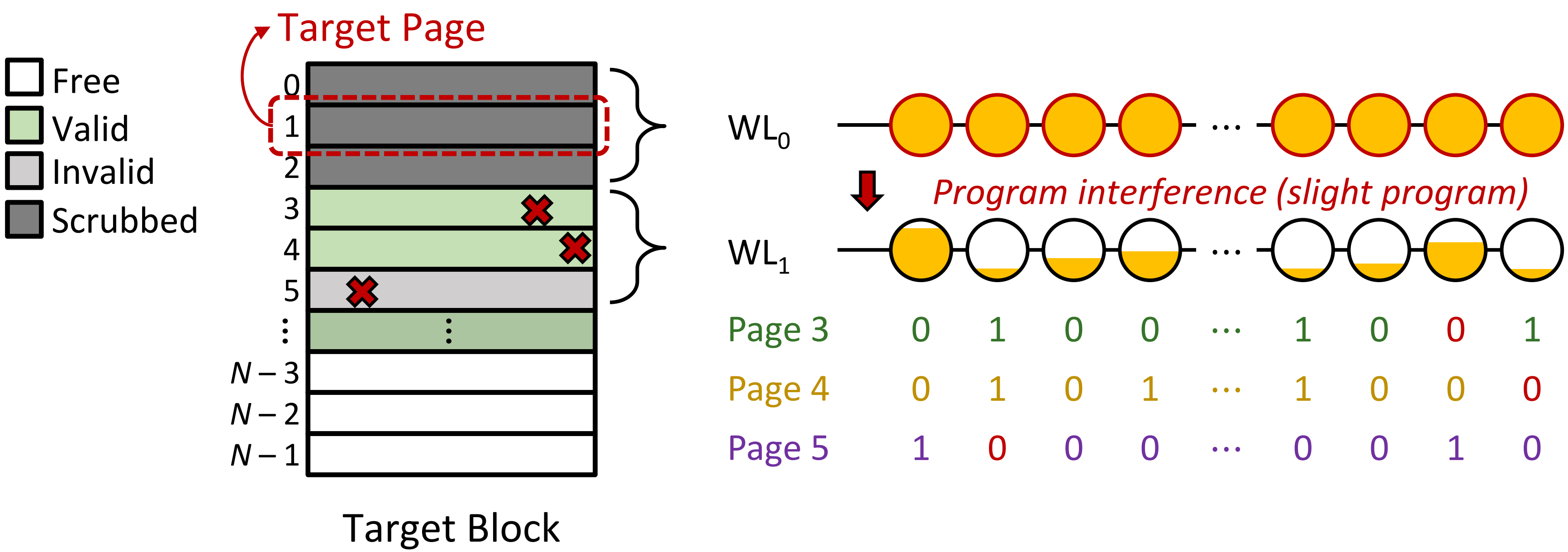
# Existing Solutions

- Scrubbing [Wei+, FAST'11]
  - Problem 1: MLC NAND flash stores multiple pages in a WL
  - Problem 2: Program interference



# Existing Solutions

- Scrubbing [Wei+, FAST'11]
  - Problem 1: MLC NAND flash stores multiple pages in a WL
  - Problem 2: Program interference




Existing solutions incur
performance, lifetime, and reliability problems
in modern NAND flash memory

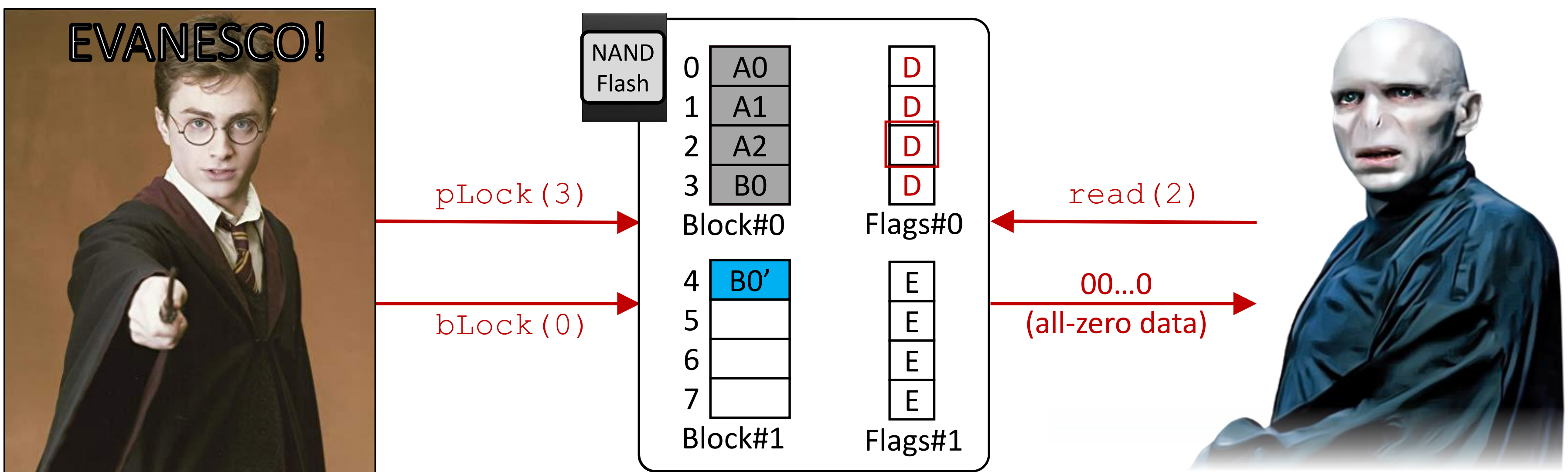
# Evanesco: Outline

- Data Remanace in NAND Flash-Based SSDs
- Evanesco: Access Control–Based Sanitization
- Evaluation Results

# Our Solution: Evanesco

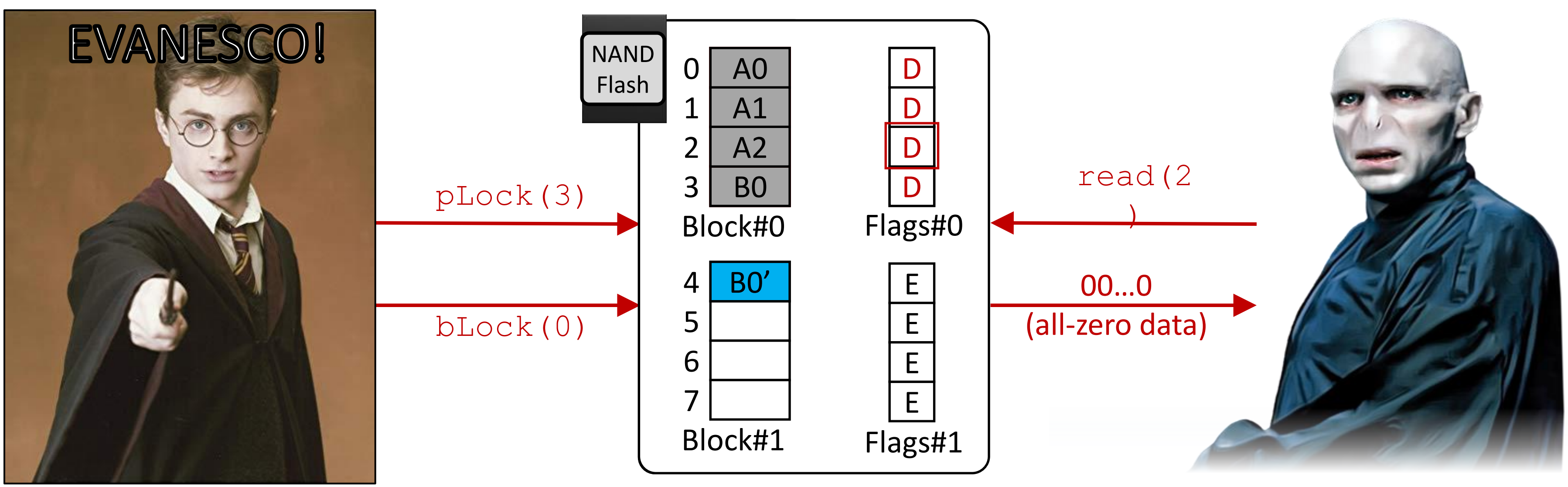
- Allows a NAND flash chip to be aware of data validity
  - On-chip access control to avoid access to invalid data
  - Low overhead: No copy operations
  - High reliability: No program interference

- Two new NAND commands: `pageLock` and `blockLock`



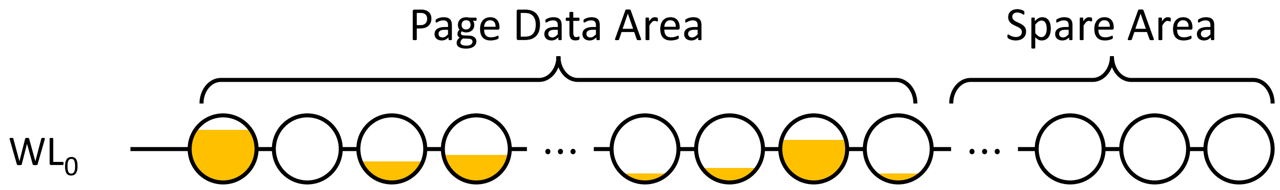
# Evanesco: Requirements

- Keep access-permission flags in a non-volatile manner
- Access-control logic inside a NAND flash chip
- Minimal area overhead → High chip densitity is paramount



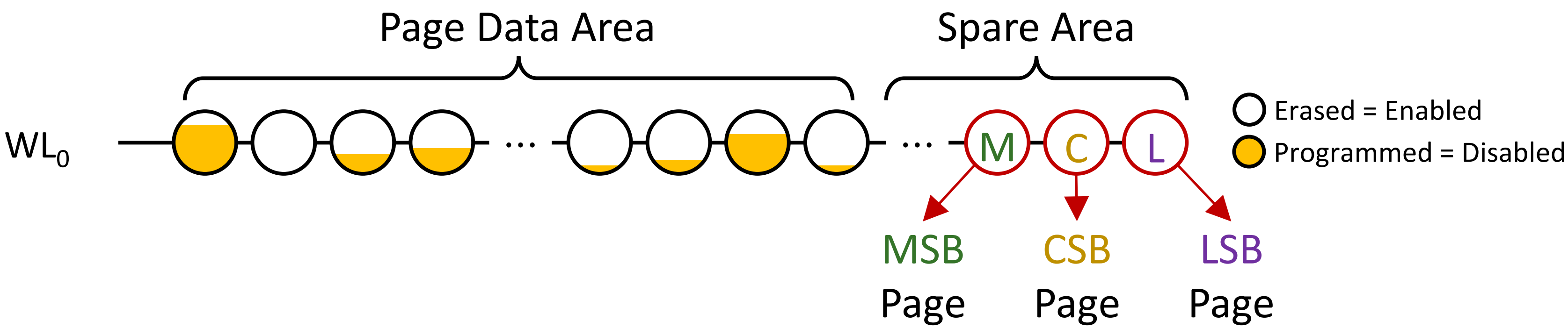
# pLock: Page-Level Data Sanitization

- On-chip access-permission flags: Spare cells in each WL



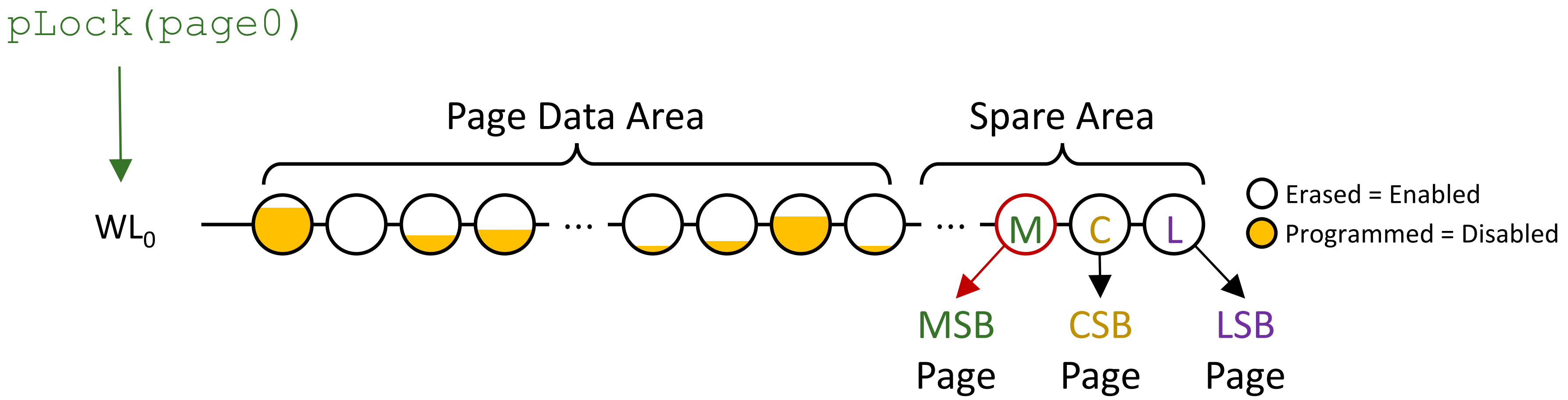
# pLock: Page-Level Data Sanitization

- On-chip access-permission flags: Spare cells in each WL



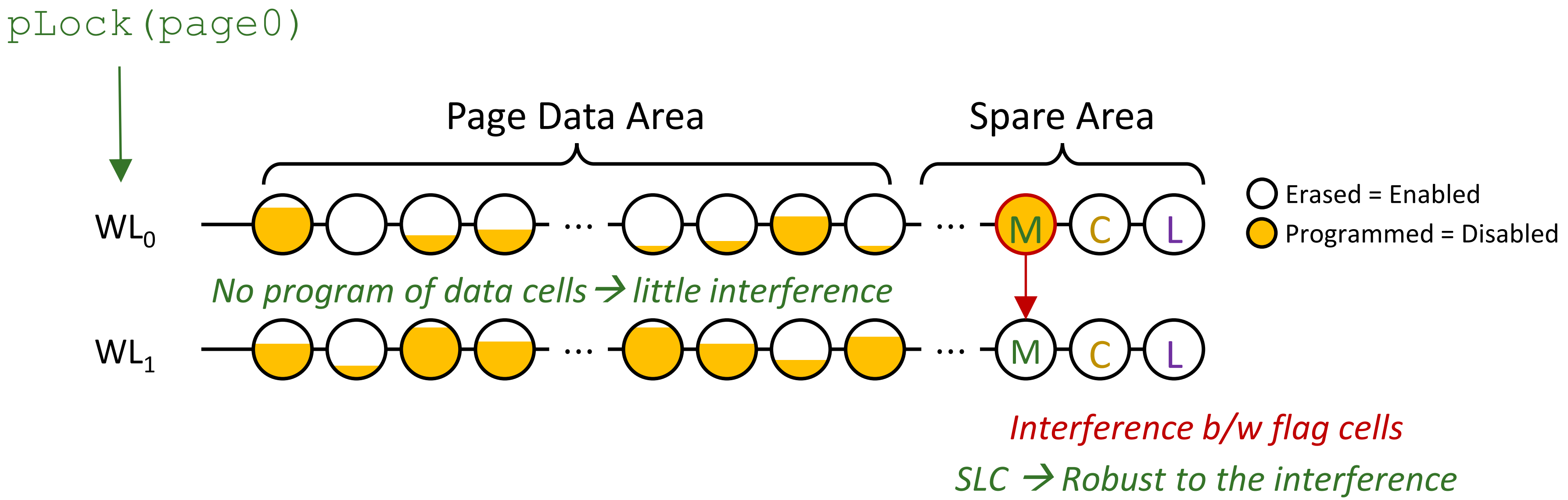
# pLock: Page-Level Data Sanitization

- On-chip access-permission flags: Spare cells in each WL



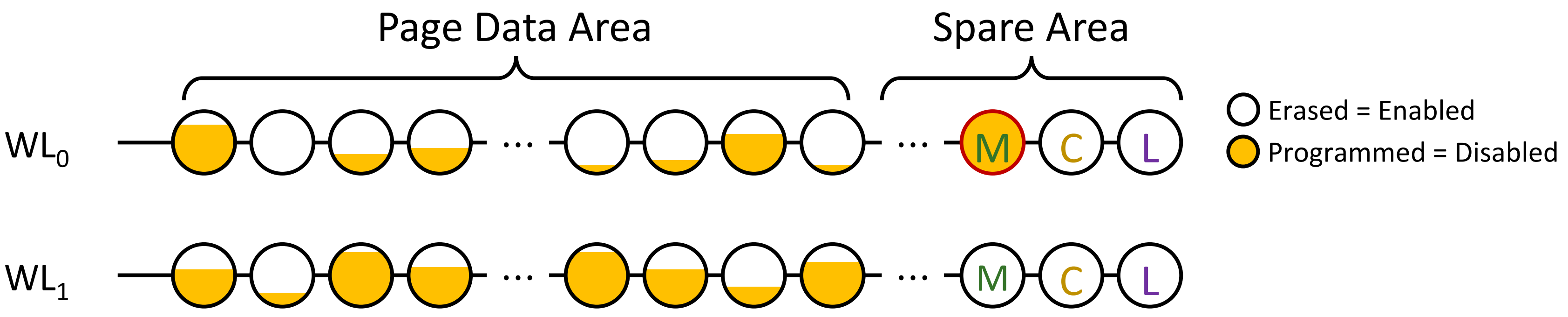
# pLock: Page-Level Data Sanitization

- On-chip access-permission flags: Spare cells in each WL



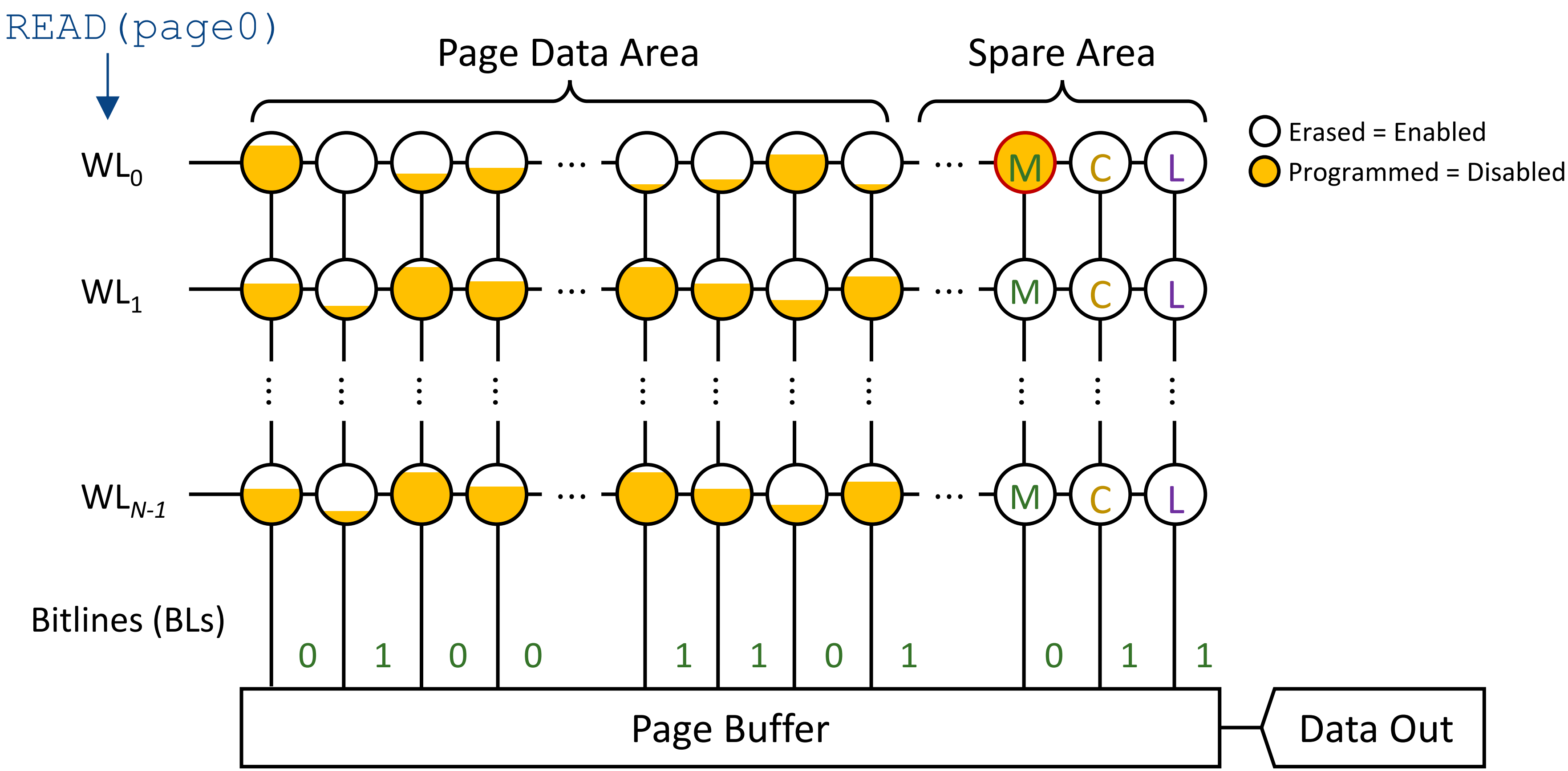
# pLock: Page-Level Data Sanitization

- On-chip access-permission flags: Spare cells in each WL
- On-chip access control logic: Small changes to data path



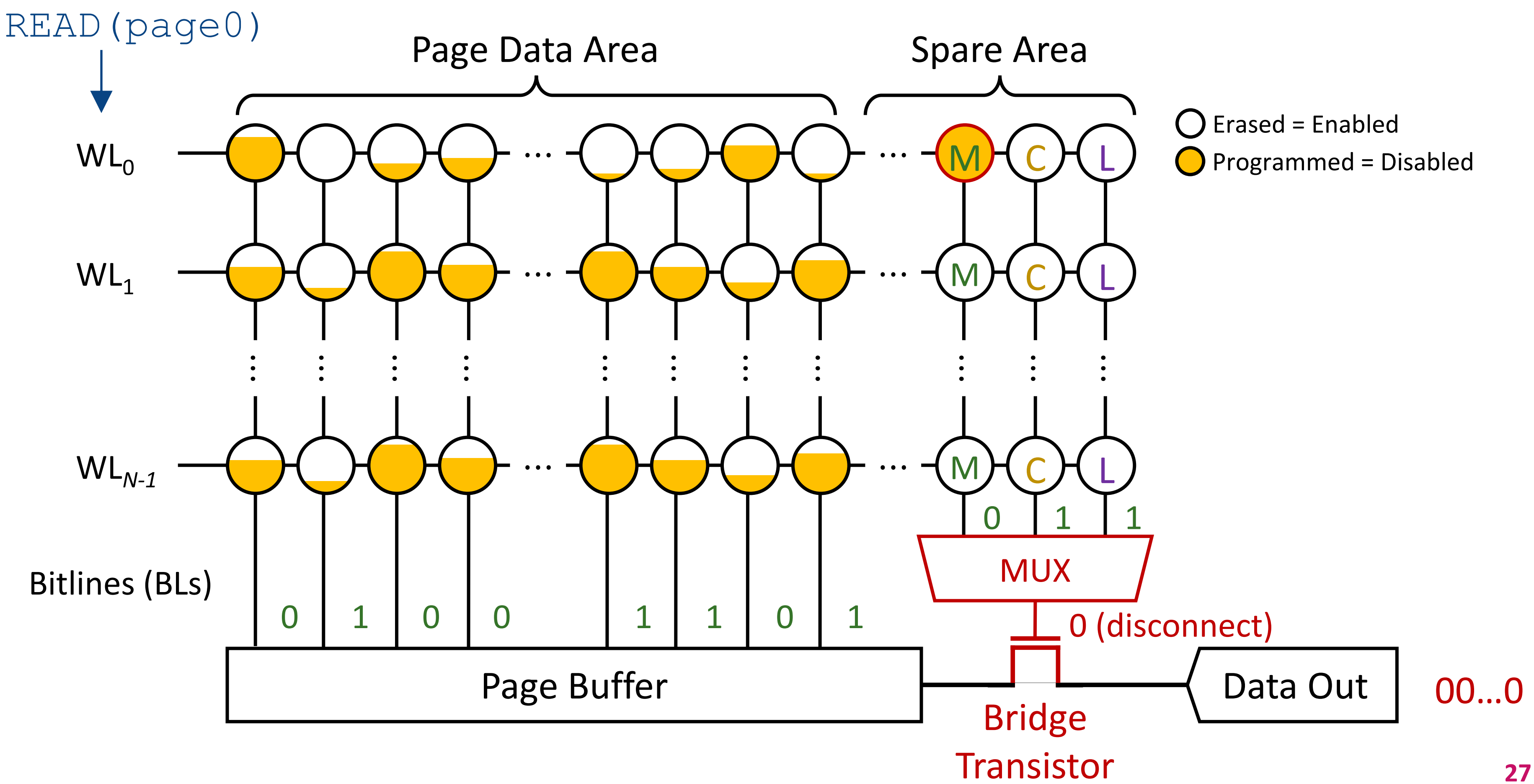
# pLock: Page-Level Data Sanitization

- On-chip access-permision flags: Spare cells in each WL
- On-chip access control logic: Small changes to data path



# pLock: Page-Level Data Sanitization

- On-chip access-permision flags: Spare cells in each WL
- On-chip access control logic: Small changes to data path`



# Real-Device Characterization

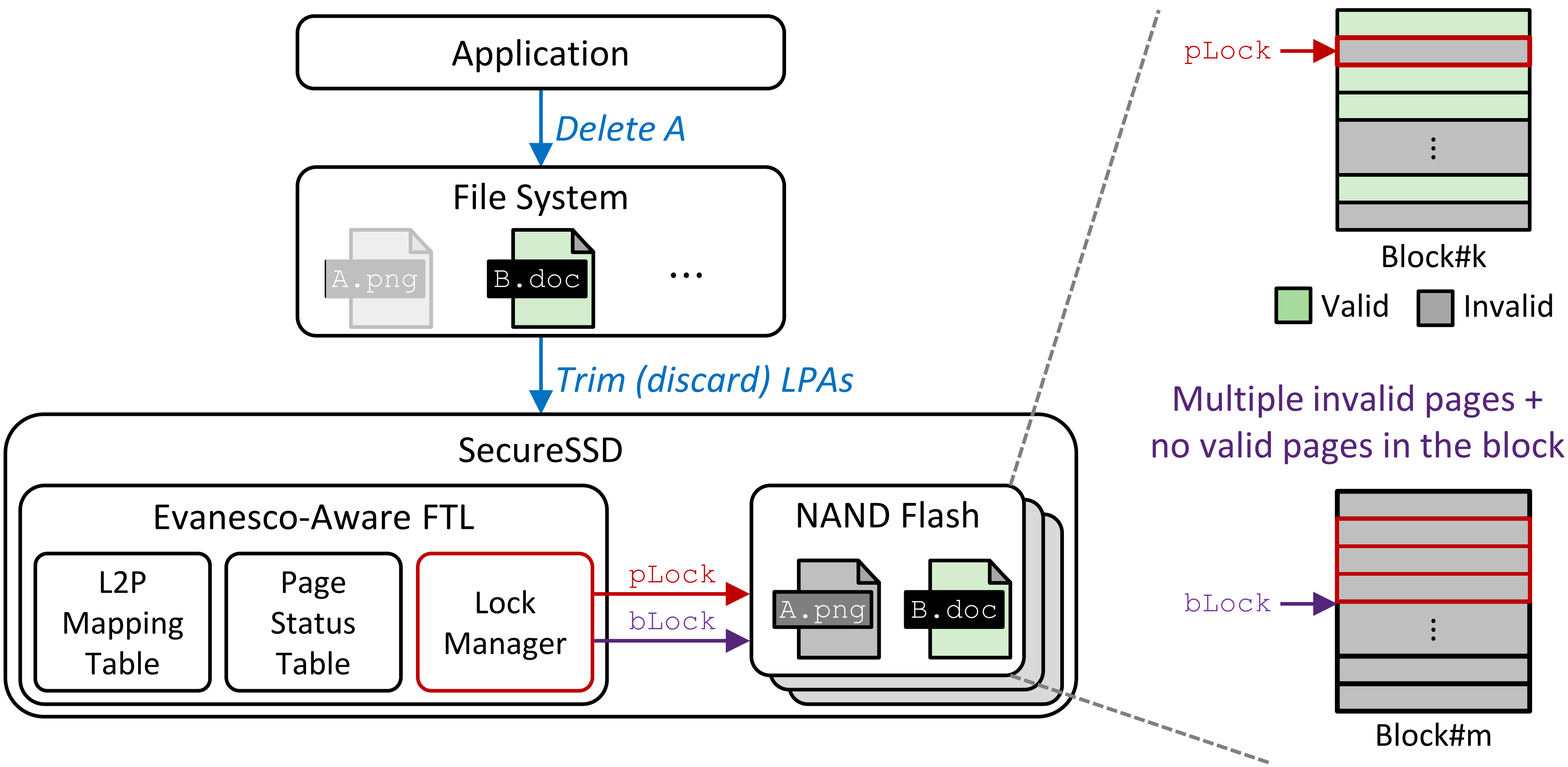
- Using 160 real 48-layer TLC NAND flash chips
- No reliability degradation for stored data
- `tPLOCK` = 100 μs, `tBLOCK` = 300 μs

Evanesco: No copy operation, no reliability issues w/ minimal changes to NAND flash chip designs

But the performance overhead is not negligible

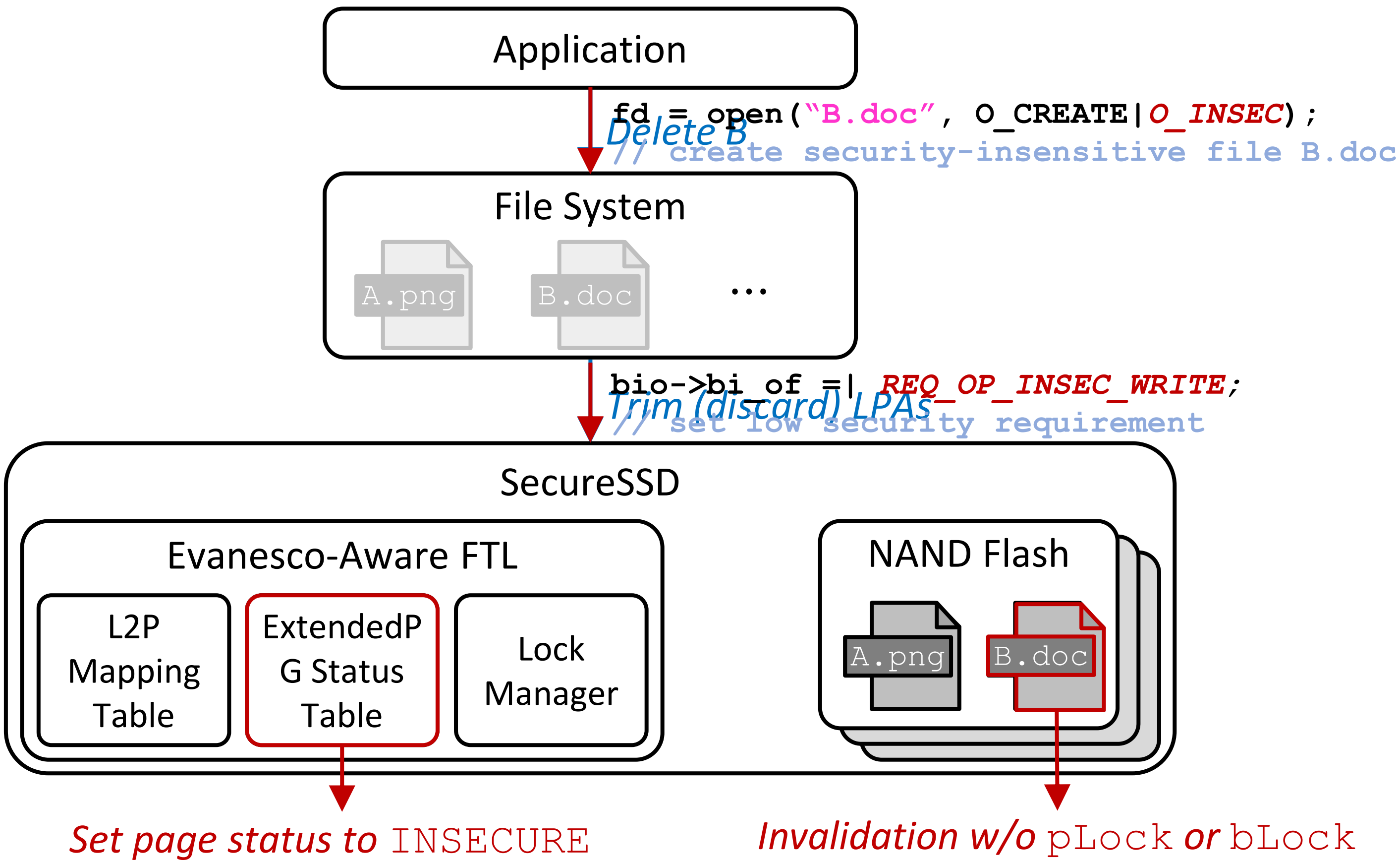
# SecureSSD: System-Level Optimization

- To minimize the performance overhead of data sanitization
  - Issues pLock and bLock commands depending on the status of the target block



# SecureSSD: System-Level Optimization

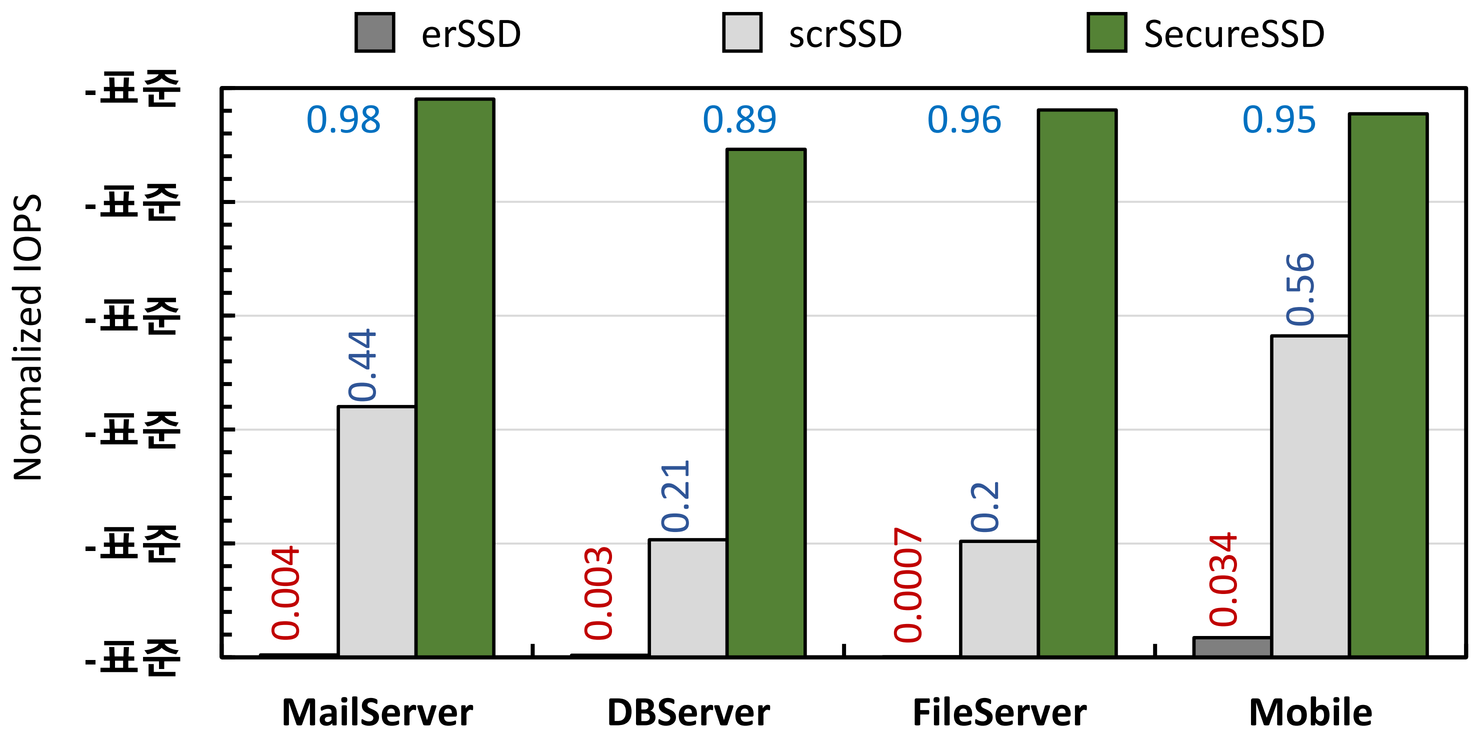
- To minimize the performance overhead of data sanitization
  - Issues pLock and bLock commands depending on the status of the target block
  - Cross-layer interactions for selective data sanitization



# Evanesco: Outline

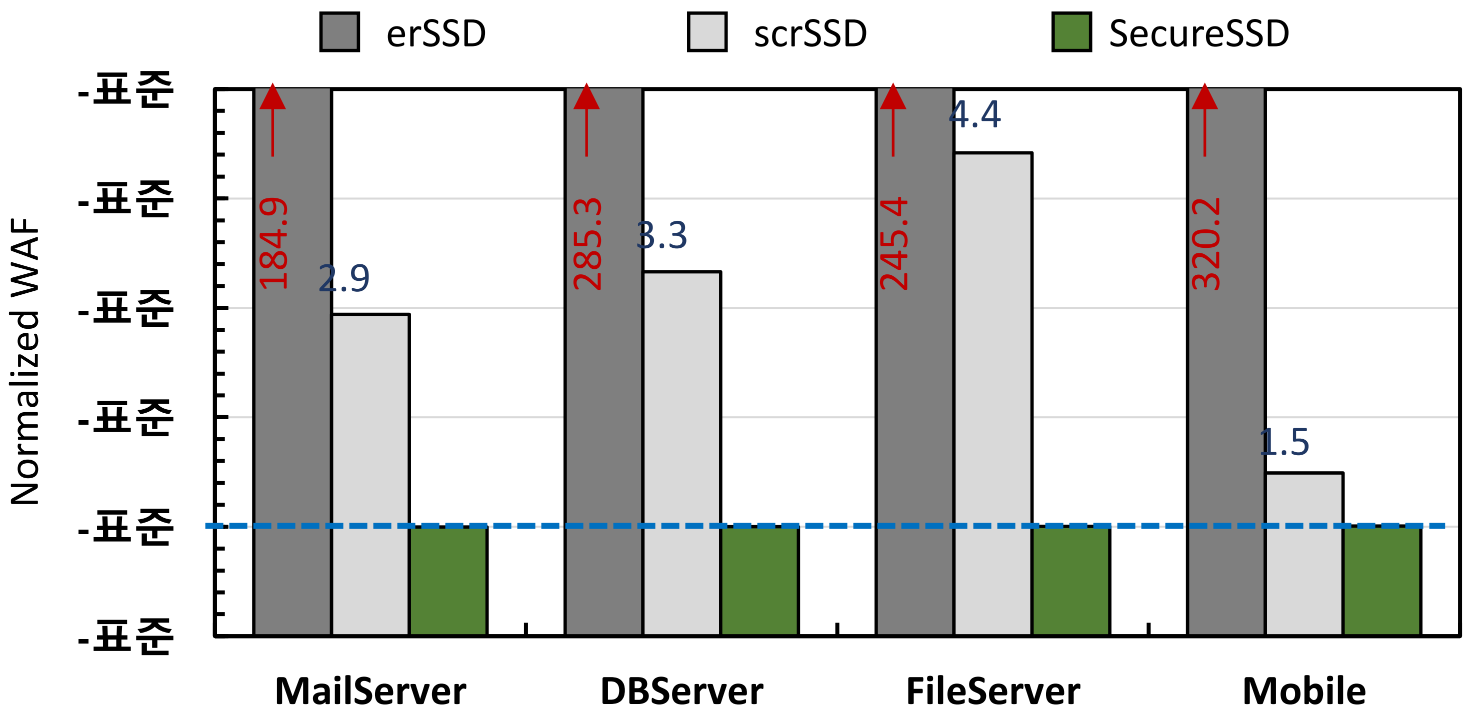
- Data Remanace in NAND Flash-Based SSDs
- Evanesco: Access Control–Based Sanitization
- Evaluation Results

# Results: Performance




Evanesco significantly reduces
performance overhead of data sanitization
(11% slowdown at most)

# Results: Lifetime




$$\textit{Write Amplification Factor}\ (WAF) = \frac{\#\ of\ physical\ pages\ written\ by\ the\ SSD}{\#\ of\ logical\ pages\ written\ by\ the\ host\ system}$$

No additional copy for data sanitization

→ No lifetime overhead

# Summary of Contribution

- Problem1: Long, non-deterministic SSD read latency
  - Due to essential reliability management (read-retry)
  - Performance degradation of data-intensive applications

- Our solution: Pipelined & adaptive read-retry
  - Leveraging device characteristics and ECC margin
  - Reducing read-retry latency
    → easy to combine with other optimizations

- Problem2: Data remanence in NAND flash-based SSDs
  - Obsolete data remains intact in SSDs for an indefinite time
  - Physical data destrubtion: prohibitive performance overheads

- Our solution: Access control-based data sanitization
  - Avoids transfer of obsolete data from NAND flash chips
  - Minimal performance and reliability overheads

# P&S Modern SSDs

## Cutting-Edge Research in SSDs

Dr. Jisung Park
Prof. Onur Mutlu
ETH Zürich
Fall 2022
04 January 2023